Advanced Scientific Computing Research
Applied Mathematics
FY 2005 Accomplishment

Research on Scientific Components at Argonne

Lois Curfman McInnes (PI), Steve Benson, Jay Larson,
Boyana Norris, and Jason Sarich
Mathematics and Computer Science Division
Argonne National Laboratory

Summary

The Center for Component Technology for Terascale Simulation Software (CCTTSS), a multi-institutional SciDAC-funded project, focuses on developing component technology for high-performance computing via the Common Component Architecture (CCA), with a goal of facilitating the interoperability and reuse of scientific software. Highlights of recent work by the Argonne contingent of the CCTTSS include developing scientific components, incorporating the CCA into climate simulations, improving the ease of use of CCA environments, and investigating issues in computational quality of service related to robust, efficient, and scalable performance.

Scientific Components. A key area of recent work among CCTTSS researchers at Argonne has been the development of production components that are used in scientific applications as well as prototype components that aid in teaching CCA concepts. Among the freely available components are tools for parallel "MxN" data redistribution, based on experience within the Model Coupling Toolkit, as well as tools for optimization and linear algebra. An important facet of this work is defining domain-specific common interfaces, which helps in realizing our vision of interchangeable scientific components. We are collaborating with the TOPS[1] SciDAC center to define common interfaces for linear and nonlinear solvers.

Optimization in Quantum Chemistry. An important goal of our project is to assure that scientific components are interoperable, able to deliver high performance, and useful to real applications. Thus, we are collaborating with chemists to perform CCA-based quantum chemistry simulations, which employ components based on the NWChem (PNNL) and MPQC (SNL) quantum chemistry codes for energy, gradient, and Hessian computations; the TaoSolver optimization component (ANL); and components based on Global Arrays (PNNL) and PETSc (ANL) for parallel linear algebra operations. Recent molecular geometry experiments have demonstrated reductions in simulation times up to 43% compared to the stand-alone chemistry packages.

Climate Components. Computational climate modeling is critical for our understanding of global processes and the consequences of human impact. Recent work at Argonne on component-based climate applications has leveraged the CCA's SIDL language-interoperability technology to allow access to the Fortran-

[1] *Terascale Optimal PDE Simulations Center*, PI: D. Keyes, http://tops-scidac.org

*Mathematics and Computer Science Division, (630) 252-5170, mcinnes@mcs.anl.gov

based Model Coupling Toolkit (MCT) from C++ and Python. We have tested the C++ bindings in a parallel coupled application that uses the MCT's MxN and parallel interpolation facilities, while collaboration with an NSF-sponsored group at the University of Chicago has resulted in a Python parallel coupled test application. This work is one facet of building a proof-of-concept component version of the Community Climate System Model (CCSM). In addition, we are developing components for post-processing and analysis of climate simulations, including a component interface to netCDF, a popular data format in the geosciences.

Usability. Argonne has taken a leading role in a new effort focusing on the usability of high-performance component technologies. We have developed a new, fully automated build system for components and are implementing an Eclipse-based development environment that aims to provide automation whenever possible, while allowing the application developer to maintain control of the software development process.

Computational Quality of Service. Component-based environments provide opportunities to improve the performance and numerical accuracy of scientific simulations. The concept of the automatic selection and configuration of components to suit a particular computational purpose is called *computational quality of service (CQoS).* As part of our CQoS research, we have developed prototype software to support adaptive algorithms in component-based applications via runtime performance monitoring and performance database manipulations (see Figure 1). We have implemented components for manipulating two types of performance databases: a low-overhead runtime database for storing and accessing information about the currently executing application instance, and another, conventional database, which is accessed

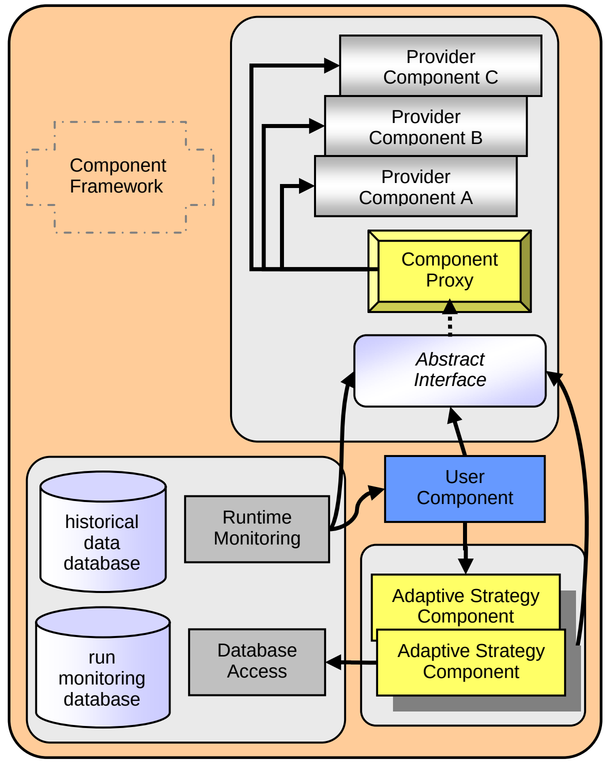

Figure 1. Computational quality of service infrastructure for adaptive solvers.

only at the beginning and end of an application to retrieve prior performance data and record any new information. The runtime database can be used for adaptive strategies, such as changing the linear solvers employed in the context of solving nonlinear partial differential equations. The larger and slower database containing information from multiple application instances is intended for offline analysis, in which large data sets can be processed and essential performance characteristics summarized.

For further information on this subject contact:

Lois Curfman McInnes (PI)
Argonne National Laboratory
Argonne, IL 60439-4844
Phone: 630-252-5170
E-mail: mcinnes@mcs.anl.gov
http://www.mcs.anl.gov/cca